Subject: IDL_IDLBridge GetVar vs. Shared Memory
Posted by Dick Jackson on Tue, 05 Dec 2006 21:27:26 GMT
View Forum Message <> Reply to Message

Hi all,

Using IDL_IDLBridge has been very helpful in some of our work lately, helping us to get more processing done while one IDL process is waiting on a DLL, and other fun. I thought I'd share some findings that show how results can come back 4-10 times faster using shared memory (SHMMAP, etc.) rather than GetVar(). Any comments on this are very welcome, and I'd be interested to see how this plays out on other hardware.

Notes:

I split out the SHM Cleanup timing since some applications may not require cleanup between successive calls.

I thought it a bit odd to have to do this when assigning the value on the bridge process:
  oBridge -> Execute, "shmA[0,0,0,0] = a"
... but if you use [0] then only the first row of values is filled in. Sure, it's faster, but... :-) (and if you use [0,0] only the first 2D plane is filled in, etc.) Here's my !Version:

{ x86 Win32 Windows Microsoft Windows 6.3 Mar 23 2006 32 64}

I'll send another posting with the .pro file as an attachment, in case that's more convenient for some to use.

Cheers,
-Dick

--
Dick Jackson Software Consulting http://www.d-jackson.com
Victoria, BC, Canada +1-250-220-6117 dick@d-jackson.com

=====

```
PRO SHMvsGetVar

;;  Using an IDL_IDLBridge process, test timings of getting results back
;;  using shared memory (SHM) vs. oBridge -> GetVar()
;;
;;  SHM appears to be four to ten times faster in my testing.
;;  Dick Jackson - dick@d-jackson.com
```

```
oBridge = Obj_New('IDL_IDLBridge')
shmName = 'SHMvsGetVar'

;;    Describe sizes of arrays to use for testing (as strings)

sizes = ['10', '1E7', '[10,1E6]', '[1E6,10]', '[1E4,1E3]', '[1E2,1E2,1E3]']
;;  '1E8','[10,1E7]','[1E7,10]','[1E4,1E4]','[1E3,1E2,1E3]','[1E 2,1E2,1E2,1E2]']

nTests = N_Elements(sizes)

FOR testI=0, nTests-1 DO BEGIN

   sizeStr = sizes[testI]
   ok = Execute('sizeNum = Long('+sizeStr+')')
   IF ~ok THEN Message, 'Execute() failed: check sizeStr "'+sizeStr+""

   Print, 'Testing with byte array of size: '+sizeStr
   oBridge -> Execute, 'a=BIndGen('+sizeStr+')'
   localA = BIndGen(sizeNum)

   ;;    Test SHM method

   t0 = SysTime(/Seconds)
   oBridge -> Execute, "SHMMap,'"+shmName+"',/Byte,Dimension="+sizeStr
   oBridge -> Execute, "shmA = SHMVar('"+shmName+"')"
   oBridge -> Execute, "shmA[0,0,0,0] = a" ; Must have N_Dims(a) or more (bug?)
   SHMMap, shmName,/Byte,Dimension=sizeNum
   shmA = SHMVar(shmName)
   shmTime = SysTime(/Seconds)-t0
   Print, Format="('    SHM:        ',F0.3,' s')",shmTime

   ;;    Check result
   IF ~(Array_Equal(Size(shmA), Size(localA)) && $
        Array_Equal(shmA, localA)) THEN Print, '   *** Result check failed!'

   ;;    Remove references to mapped variables and unmap memory segments
   t0 = SysTime(/Seconds)
   shmA = 0B
   SHMUnmap, shmName
   oBridge -> Execute, "shmA = 0B"
   oBridge -> Execute, "SHMUnmap,'"+shmName+"'"
   Print, Format="('    SHM+Cleanup:',F0.3,' s')",shmTime+(SysTime(/Seconds)-t0)

   ;;    Test GetVar method

   t0 = SysTime(/Seconds)
   getVarA = oBridge -> GetVar('a')
   Print, Format="('    GetVar:     ',F0.3,' s')",SysTime(/Seconds)-t0
```

```
  ;;    Check result
  IF ~(Array_Equal(Size(getVarA), Size(localA)) && $
       Array_Equal(getVarA, localA)) THEN Print, '   *** Result check failed!'

  Wait, 0.001                    ; To allow Print statements to flush

ENDFOR

Obj_Destroy, oBridge

END
```